Modeling and Performance Evaluation of Computer Systems Security Operation[1]

D. Guster[2]

St.Cloud State University[3]

N.K. Krivulin[4]

St.Petersburg State University[5]

Abstract

A model of computer system security operation is developed based on the fork-join queueing network formalism. We introduce a security operation performance measure, and show how it may be used to performance evaluation of actual systems.

Keywords: computer system security, security attack, security vulnerability, performance evaluation, fork-join queueing networks

1 Introduction

The explosive growth in computer systems and networks has increased the role of computer security within organizations [4]. In many cases, ineffective protection against computer security treats leads to considerable damage, and even can cause an organization to be paralized. Therefore, the development of new models and methods of performance analysis of security systems seems to be very important.

In this paper, we propose a model of computer security operation, and introduce its related performance measure. It is shown how the model can be applied to performance evaluation of actual systems. Finally, a technique of security system performance analysis is described and its practical implementation is discussed.

We conclude with an appendix which contains technical details concerning fork-join network representation of the model, and related results.

[1]The work was partially supported by the Russian Foundation for Basic Research, Grant #00-01-00760.

[2]E:-mail:Guster@mcs.stcloudstate.edu

[3]Department of Statistics, 720 4th Ave. S., St.Cloud, MN 56301-4442

[4]E-mail: Nikolai.Krivulin@pobox.spbu.ru

[5]Bibliotechnaya Sq. 2, Petrodvorets, 198904 St.Petersburg, Russia

2 A Security Operation Model

In this paper, we deal with the current security activities (see Fig. 1) that mainly relate to the actual security threats rather than to strategic or long-term issues of security management.

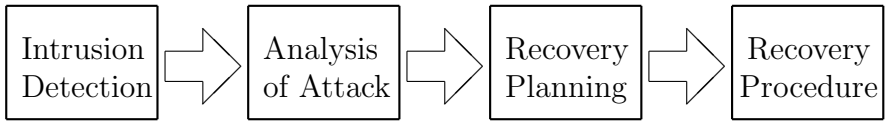


Figure 1. Computer systems security activities.

Consider the model of security operation in an organization, presented in Fig. 2. Each operational cycle starts with security attack detection based on audit records and system/errors log analysis, traffic analysis, or user reports. In order to detect an intrusion, automated tools of security monitoring are normally used including procedures of statistical anomaly detection, rule-based detection, and data integrity control [4].

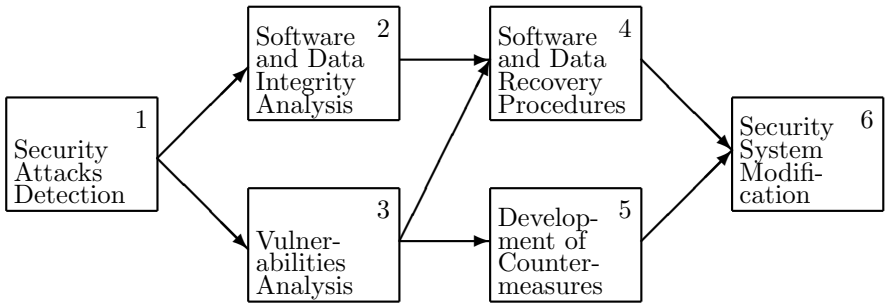


Figure 2. A security analysis and maintenance model.

After security attack detection and identification, the integrity of system/application software and data in storage devices has to be examined to search for possible unauthorized modifications or damages made by the intruder. The investigation procedure can exploit file lists and checksum analysis, hash functions, and other automated techniques.

In parallel, the system vulnerabilities, which allow the intruder to attack, should be identified and investigated. The vulnerability analysis normally presents an informal procedure, and therefore, it can hardly be performed automatically.

Based on the results of integrity analysis, a software and data recovery procedure can be initiated using back-up servers and reserving storage devices. It has to take into account the security vulnerabilities identified at the previous step, so as to provide for further improvements in the entire security system.

Along with the recovery procedure, the development of a complete set of countermeasures against similar attacks should be performed. Finally, the operational

cycle is concluded with appropriate modifications of software, data bases, and system security policies and procedures.

We assume that the organization has appropriate personnel integrated in a Computer Emergency Response Team, available to handle the attack. The team would include at least two subteams working in parallel, one to perform integrity analysis and recovery procedures, and another to do vulnerability analysis and development of countermeasures. At any time instant, each subteam can deal with only one security incident. Any procedure may be started as soon as all prior procedures according to the model in Fig. 2, have been completed. If a request to handle a new incident occurs when a subteam is still working on a procedure, the request has to wait until the processing of that procedure is completed.

We denote by τ_{1k} a random variable (r.v.) that represents the time interval between detections of the kth attack and its predecessor. Furthermore, we introduce r.v.'s $\tau_{ik},\ i = 2, \ldots, 6$, to describe the time of the kth instant of procedure i in the model. We assume $\tau_{i1}, \tau_{i2}, \ldots$, to be independent and identically distributed (i.i.d.) r.v.'s with finite mean and variance for each $i,\ i = 1, \ldots, 6$. At the same time, we do not require of independence of $\tau_{1k}, \ldots, \tau_{6k}$ for each $k,\ k = 1, 2, \ldots$.

3 Security Operation Performance Evaluation

In order to describe system performance, we introduce the following notations. Let $\overline{T}_A$ be the mean time between consecutive security attacks (the attack cycle time), and $\overline{T}_S$ be the mean time required to completely handle an attack (the recovery cycle time), as the number of attacks k tends to ∞.

In devising the security operation performance measure, one can take the ratio

$$R = \overline{T}_S / \overline{T}_A.$$

With the natural condition $\overline{T}_S \leq \overline{T}_A$, one can consider R as the time portion the system is under recovery, assuming $k \to \infty$.

First note that the attack cycle time can immediately be evaluated as the mean value: $\overline{T}_A = \mathrm{E}[\tau_{11}]$.

Now consider the cycle time of the entire system, which can be defined as the mean time interval between successive completions of security system modification procedures as the number of attacks $k \to \infty$. As one can prove (see Appendix for further details), the system cycle time γ can be calculated as

$$\gamma = \max\{\mathrm{E}[\tau_{11}], \ldots, \mathrm{E}[\tau_{61}]\}.$$

In order to evaluate the recovery cycle time, we assume the system will operate under the maximum traffic level, which can be achieved when all the time intervals between attacks are set to 0. Clearly, under that condition, the system cycle time can be taken as a reasonable estimate of the recovery cycle time.

Considering that now $\mathrm{E}[\tau_{11}] = 0$, we get the recovery cycle time in the form

$$\overline{T}_S = \max\{\mathrm{E}[\tau_{21}], \ldots, \mathrm{E}[\tau_{61}]\}.$$

4 Performance Analysis and Discussion

In fact, the above model presents a quite simple but useful tool for security system operation management. It may be used to make decision on the basis of a few natural parameters of the security operation process.

Let us represent the ratio R in the form

$$R = \max\{\mathrm{E}[\tau_{21}], \ldots, \mathrm{E}[\tau_{61}]\}/\mathrm{E}[\tau_{11}],$$

and assume the attack rate determined by $\mathrm{E}[\tau_{11}]$, to be fixed.

Taking into account that the above result has been obtained based on the assumption of an infinite number of attacks, we arrive at the following conclusion. As the number of attacks becomes sufficiently large, the performance of the system is determined by the time of the longest procedure involved in the system operation, whereas the impact of the order of performing the procedures disappears.

It is clear that in order to improve system performance, the system security manager (administrator) should first concentrate on decreasing the mean time required to perform the longest procedure within the security operation model, then consider the second longest procedure, and so on. The goal of decreasing the time can be achieved through partition of a whole procedure into subprocedures, which can be performed in parallel, or through rescheduling of the entire process with redistribution of particular activities between procedures.

In practice, the above model and its related ratio R can serve as the basis for efficient monitorization of organizational security systems. Because the introduction of new countermeasures may change the attack cycle time, the monitoring requires updating this parameter after each modification of the system.

Finally note, the above model can be easily extended to cover security operational processes, which consist of different procedures and precedence constraints.

Appendix

In order to describe the above security system operational model in a formal way, we exploit the fork-join network formalism proposed in [1].

The fork-join networks present a class of queueing systems, which allow for splitting a customer into several new customers at one node, and of merging customers into one at another node. In order to represent the dynamics of such networks, we use a $(\max, +)$-algebra based approach developed in [2].

The $(\max, +)$-algebra is a triple $\langle R_\varepsilon, \oplus, \otimes\rangle$, where $R_\varepsilon = R \cup \{\varepsilon\}$ with $\varepsilon = -\infty$. The operations $\oplus$ and $\otimes$ are defined for all $x, y \in R_\varepsilon$ as

$$x \oplus y = \max(x, y), \quad x \otimes y = x + y.$$

The $(\max, +)$-algebra of matrices is introduced in the ordinary way with the matrix $\mathcal{E}$ with all its entries equal ε, taken as the null matrix, and the matrix $E = \mathrm{diag}(0, \ldots, 0)$ with its off-diagonal entries equal ε, as the identity.

We introduce the vector $\boldsymbol{x}(k) = (x_1(k), \ldots, x_n(k))^T$ as the kth service completion times at the network nodes, and the diagonal matrix $\mathcal{T}_k = \mathrm{diag}(\tau_{1k}, \ldots, \tau_{nk})$

with given nonnegative random variables τ_{ik} representing the kth service time at node i, $i = 1, \ldots, n$, and the off-diagonal entries equal ε.

The dynamics of acyclic fork-join networks can be described by the stochastic difference equation (see [2] for further details)

$$\boldsymbol{x}(k) = A(k) \otimes \boldsymbol{x}(k-1), \qquad A(k) = \bigoplus_{j=0}^{p} (\mathcal{T}_k \otimes G^T)^j \otimes \mathcal{T}_k, \tag{1}$$

where G is a matrix with the elements

$$g_{ij} = \begin{cases} 0, & \text{if there exists arc } (i,j) \text{ in the network graph,} \\ \varepsilon, & \text{otherwise,} \end{cases}$$

and p is the length of the longest path in the graph.

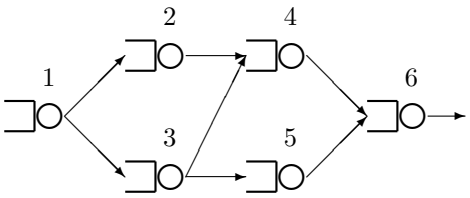
The matrix G is normally referred to as the support matrix of the network. Note that since the network graph is acyclic, we have $G^q = \mathcal{E}$ for all $q > p$.

The cycle time of the network is defined as

$$\gamma = \lim_{k \to \infty} \|\boldsymbol{x}(k)\|,$$

where $\|\boldsymbol{x}(k)\| = \max_i x_i(k)$. Clearly, if this limit exists, it can be found as $\lim_{k\to\infty} \|A_k\|$, where $A_k = A(k) \otimes \cdots \otimes A(1)$.

As it is easy to see, the fork-join network representation of the above security operation model takes the form presented in Fig. 3.



(a) Network scheme

$$G = \begin{pmatrix} \varepsilon & 0 & 0 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & 0 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & 0 & 0 & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{pmatrix}$$

(b) Support matrix

Figure 3. The fork-join queueing network model.

For the network graph, we have $p = 3$. Therefore, we get equation (1) with $A(k) = (E \oplus \mathcal{T}_k \otimes G^T \oplus (\mathcal{T}_k \otimes G^T)^2) \oplus (\mathcal{T}_k \otimes G^T)^3) \otimes \mathcal{T}_k$.

Let us consider an arbitrary fork-join queueing network with n nodes, which is governed by equation (1). We assume that the matrix G at (1) has the upper triangular form. Since the network graph is acyclic, the network nodes can always be renumbered so that the matrix G become upper triangular.

Now we describe a tandem queueing system associated with the above network. We assume the evolution of the tandem system to be governed by the equation

$$\boldsymbol{x}(k) = B(k) \otimes \boldsymbol{x}(k-1), \qquad B(k) = \bigoplus_{j=0}^{n} (\mathcal{T}_k \otimes H^T)^j \otimes \mathcal{T}_k,$$

where H is a support matrix with the elements

$$h_{ij} = \begin{cases} 0, & \text{if } i+1=j, \\ \varepsilon, & \text{otherwise.} \end{cases}$$

Note that both matrices $A(k)$ and $B(k)$ are determined by the common matrix $\mathcal{T}_k$, but different support matrices G and H. Clearly, the longest path in the graph associated with the tandem queue is assumed to be equal n.

Lemma 1. *For all $k = 1, 2, \ldots$, it holds that $A(k) \leq B(k)$.*

Proof: As it is easy to verify, for any integer $q > 0$, it holds

$$G^q \leq H \oplus H^2 \oplus \cdots \oplus H^n.$$

Furthermore, since $\mathcal{T}_k$ has only nonnegative entries on the diagonal, we have for any $q > 1$,

$$H^q \otimes \mathcal{T}_k \leq (H \otimes \mathcal{T}_k)^q.$$

By applying the above inequalities together with the condition that $H^m = \mathcal{E}$ for all $m > n$, we arrive at the inequality

$$(G \otimes \mathcal{T}_k)^q \leq (H \otimes \mathcal{T}_k) \oplus (H \otimes \mathcal{T}_k)^2 \oplus \cdots \oplus (H \otimes \mathcal{T}_k)^n.$$

Taking into account that the last inequality is valid for all $q > 0$, we have

$$\mathcal{T}_k \otimes \bigoplus_{j=0}^{p} (G \otimes \mathcal{T}_k)^j \leq \mathcal{T}_k \otimes \bigoplus_{j=0}^{n} (H \otimes \mathcal{T}_k)^j.$$

It remains to transpose the both side of the inequality to get the desired result.

By applying the above lemma together with the result in [3], one can prove the following statement.

Lemma 2. *Suppose that for the acyclic fork-join queueing network, the random variables $\tau_{i1}, \tau_{i2}, \ldots$, are i.i.d. for each $i = 1, \ldots, n$ with finite mean $\mathrm{E}[\tau_{i1}] \geq 0$ and variance $\mathrm{D}[\tau_{i1}]$. Then the cycle time γ can be evaluated as*

$$\gamma = \max\{\mathrm{E}[\tau_{11}], \ldots, \mathrm{E}[\tau_{n1}]\}.$$

References

1. F. Baccelli and A.M. Makowski, Queueing Models for Systems with Synchronization Constraints, *Proceedings of the IEEE*, Vol.77, No.1, 1989, pp.138-160.

2. N.K. Krivulin, Algebraic Modeling and Performance Evaluation of Acyclic Fork-Join Queueing Networks, *Advances in Stochastic Simulation Methods, Statistics for Industry and Technology*, (N. Balakrishnan, V. Melas, S. Ermakov, Eds.), Birkhäuser, Boston, 2000, pp.63-81.

3. N.K. Krivulin and V.B. Nevzorov, Evaluation of the Mean Interdeparture Time in Tandem Queueing Systems, *This proceedings*, 2001.

4. W. Stallings, *Network and Internetwork Security: Principles and Practice*, Prentice-Hall, Englewood Cliffs, 1995.